

LAMPIRAN A KODE PROGRAM

A.1 KODE PROGRAM SENSOR / AKTUATOR

SmartBag.ino

Kode Program Perangkat Keras Sensor dan Aktuator	
1	/*
2	* Created by Muhammad Yaqub
3	*/
4	#include <TimeLib.h>
5	#include <ESP8266WiFi.h>
6	#include <WiFiUdp.h>
7	#include <SPI.h>
8	#include <MFRC522.h>
9	#include <Wire.h>
10	#include <FirebaseArduino.h>
11	#include "SoftwareSerial.h"
12	
13	// Firebase Setup
14	#define FIREBASE_HOST "smartbag-b64b8.firebaseio.com"
15	#define FIREBASE_AUTH "idX9y5iDKOfRwrMVKghhHgAmlykPywDYw9MjPsOa"
16	
17	//RFID Reader Pin
18	#define SS_PIN D4 //D2
19	#define RST_PIN D3 //D1
20	
21	// Wifi
22	const char* ssid = "Theex-HQ";
23	//"Theex-HQ"; "FeelsBadMan"; "Lilian";
24	const char* password = "JuraganPecelYeye";
25	//"JuraganPecelYeye"; "jangkrik39A"; "qwertyui";
26	
27	// NTP Servers:
28	static const char ntpServerName[] = "us.pool.ntp.org";
29	//static const char ntpServerName[] = "time.nist.gov";
30	time_t prevDisplay = 0; // when the digital clock was displayed
31	const int timeZone = 7; // +7 GMT Waktu Indonesia Barat
32	
33	WiFiUDP Udp;
34	unsigned int localPort = 8888; // local port to listen for UDP
35	packets
36	
37	time_t getNtpTime();
38	void digitalClockDisplay();
39	void printDigits(int digits);
40	void sendNTPpacket(IPAddress &address);
41	

Kode Program Perangkat Keras Sensor dan Aktuator	
42	// MPU6050 Slave Device Address
43	const uint8_t MPU6050SlaveAddress = 0x68;
44	
45	// Select SDA and SCL pins for I2C communication
46	const uint8_t scl = D1;
47	const uint8_t sda = D2;
48	
49	// sensitivity scale factor respective to full scale setting
50	provided in datasheet
51	const uint16_t AccelScaleFactor = 16384;
52	const uint16_t GyroScaleFactor = 131;
53	
54	// MPU6050 few configuration register addresses
55	const uint8_t MPU6050_REGISTER_SMPLRT_DIV = 0x19;
56	const uint8_t MPU6050_REGISTER_USER_CTRL = 0x6A;
57	const uint8_t MPU6050_REGISTER_PWR_MGMT_1 = 0x6B;
58	const uint8_t MPU6050_REGISTER_PWR_MGMT_2 = 0x6C;
59	const uint8_t MPU6050_REGISTER_CONFIG = 0x1A;
60	const uint8_t MPU6050_REGISTER_GYRO_CONFIG = 0x1B;
61	const uint8_t MPU6050_REGISTER_ACCEL_CONFIG = 0x1C;
62	const uint8_t MPU6050_REGISTER_FIFO_EN = 0x23;
63	const uint8_t MPU6050_REGISTER_INT_ENABLE = 0x38;
64	const uint8_t MPU6050_REGISTER_ACCEL_XOUT_H = 0x3B;
65	const uint8_t MPU6050_REGISTER_SIGNAL_PATH_RESET = 0x68;
66	
67	int16_t AccelX, AccelY, AccelZ, Temperature, GyroX, GyroY, GyroZ;
68	int notag = 0;
69	bool inventoryIn;
70	
71	
72	
73	MFRC522 mfrc522(SS_PIN, RST_PIN); // Create MFRC522 instance.
74	
75	void WiFiEvent(WiFiEvent_t event) {
76	Serial.printf("[WiFi-event] event: %d\n", event);
77	
78	switch (event) {
79	case WIFI_EVENT_STAMODE_GOT_IP:
80	Serial.println("WiFi connected");
81	Serial.println("IP address: ");
82	Serial.println(WiFi.localIP());
83	
84	break;
85	case WIFI_EVENT_STAMODE_DISCONNECTED:
86	Serial.println("WiFi lost connection");
87	break;

Kode Program Perangkat Keras Sensor dan Aktuator	
88	}
89	}
90	
91	void setup()
92	{
93	
94	Serial.begin(9600); // Initiate a serial communication
95	while (!Serial) ; // Needed for Leonardo only
96	
97	SPI.begin(); // Initiate SPI bus
98	mfr522.PCD_Init(); // Initiate MFRC522
99	Serial.println();
100	Serial.println("RFID Reader OK.");
101	
102	Wire.begin(sda, scl);
103	MPU6050_Init();
104	// delete old config
105	WiFi.disconnect(true);
106	
107	delay(100);
108	
109	WiFi.onEvent(WiFiEvent);
110	
111	WiFi.begin(ssid, password);
112	while (WiFi.status() != WL_CONNECTED) {
113	delay(500);
114	Serial.print(".");
115	}
116	Serial.println("Starting UDP");
117	Udp.begin(localPort);
118	Serial.print("Local port: ");
119	Serial.println(Udp.localPort());
120	Serial.println("waiting for sync");
121	setSyncProvider(getNtpTime);
122	setSyncInterval(300);
123	
124	Serial.println();
125	Serial.println();
126	Serial.println("Wait for WiFi... ");
127	delay(100);
128	
129	pinMode(D0, OUTPUT);
130	pinMode(D8, OUTPUT);
131	Serial.println();
132	Serial.println("LED OK.");
133	Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);

Kode Program Perangkat Keras Sensor dan Aktuator	
134	Firebase.stream("/");
135	Serial.println();
136	Serial.println("Firebase Stream OK.");
137	}
138	void loop()
139	{
140	digitalWrite(D0, LOW);
141	digitalWrite(D8, LOW);
142	if (timeStatus() != timeNotSet) {
143	if (now() != prevDisplay) { //update the display only if time
144	has changed
145	prevDisplay = now();
146	digitalClockDisplay();
147	}
148	}
149	
150	readAccel();
151	delay(500);
152	scanRFID();
153	
154	}
155	double AxPrev, AyPrev, AzPrev, AxNext, AyNext, AzNext, rangeAz;
156	int counterAccel = 0;
157	int counterIdle = 0;
158	
159	void readAccel(){
160	double Ax, Ay, Az;
161	Read_RawValue(MPU6050SlaveAddress,
162	MPU6050_REGISTER_ACCEL_XOUT_H);
163	
164	//divide each with their sensitivity scale factor
165	Ax = (double)AccelX / AccelScaleFactor;
166	Ay = (double)AccelY / AccelScaleFactor;
167	Az = (double)AccelZ / AccelScaleFactor;
168	if (counterAccel == 0){
169	AxPrev = Ax;
170	AzPrev = Az;
171	AyPrev = Ay;
172	counterAccel = 1;
173	}else if(counterAccel == 1){
174	AxNext = Ax;
175	AzNext = Az;
176	AyPrev = Ay;
177	counterAccel = 0;
178	counterIdle = counterIdle + 1;
179	rangeAz = AzPrev - AzNext;

Kode Program Perangkat Keras Sensor dan Aktuator	
180	Serial.print("Az Range : ");Serial.println(AzPrev - AzNext);
181	if (rangeAz >= 0.12 rangeAz <= -0.12){
182	Serial.print("Tas Diangkat");
183	counterIdle = 0;
184	if (Firebase.getInt("tasMove") != 1){
185	Firebase.set("tasMove", 1);
186	}
187	}else if (counterIdle == 10 && rangeAz <= 0.03 && rangeAz >= -
188	0.03){
189	if (Firebase.getInt("tasMove") != 0){
190	Firebase.set("tasMove", 0);
191	}
192	}
193	}
194	}
195	
196	void scanRFID(){
197	// Look for new cards
198	if (! mfrc522.PICC_IsNewCardPresent())
199	{
200	return;
201	}
202	// Select one of the cards
203	if (! mfrc522.PICC_ReadCardSerial())
204	{
205	return;
206	}
207	
208	getUIDtag();
209	
210	}
211	
212	//Show UID on serial monitor
213	void getUIDtag(){
214	Serial.println();
215	Serial.print(" UID tag :");
216	String content= "";
217	byte letter;
218	for (byte i = 0; i < mfrc522.uid.size; i++)
219	{
220	Serial.print(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " ");
221	Serial.print(mfrc522.uid.uidByte[i], HEX);
222	content.concat(String(mfrc522.uid.uidByte[i] < 0x10 ? " 0" :
223	" "));
224	content.concat(String(mfrc522.uid.uidByte[i], HEX));
225	}

Kode Program Perangkat Keras Sensor dan Aktuator	
226	content.toUpperCase();
227	Serial.println();
228	sendUIDtoFirebase(content);
229	}
230	
231	void sendUIDtoFirebase(String content){
232	String UIDTime = String(day()) + "-" + String(month()) + "-" +
233	String(year()) + " " + String(hour()) + ":" + String(minute()) +
234	":" + String(second());
235	if (Firebase.getInt("inventory/"+content.substring(1)+"/status")
236	== 0)
237	{
238	Firebase.set("inventory/"+content.substring(1)+"/status", 1);
239	
240	Firebase.setString("inventory/"+content.substring(1)+"/in/time",
241	UIDTime);
242	Serial.println("Inventory In, Status Data uploaded");
243	inventoryIn = true;
244	}else{
245	Firebase.set("inventory/"+content.substring(1)+"/status", 0);
246	
247	Firebase.setString("inventory/"+content.substring(1)+"/out/time",
248	UIDTime);
249	Serial.println("Inventory Out, Status Data uploaded");
250	inventoryIn = false;
251	}
252	statusLED();
253	notag = 1;
254	Serial.println();
255	}
256	
257	void statusLED(){
258	if(inventoryIn){
259	digitalWrite(D8, HIGH); //Lampu Hijau Menyala
260	delay(2000);
261	}else{
262	digitalWrite(D0, HIGH); //Lampu Merah Menyala
263	delay(2000);
264	}
265	}
266	
267	void I2C_Write(uint8_t deviceAddress, uint8_t regAddress, uint8_t
268	data) {
269	Wire.beginTransmission(deviceAddress);
270	Wire.write(regAddress);
271	Wire.write(data);

Kode Program Perangkat Keras Sensor dan Aktuator	
272	Wire.endTransmission();
273	}
274	
275	// read all 14 register
276	void Read_RawValue(uint8_t deviceAddress, uint8_t regAddress) {
277	Wire.beginTransmission(deviceAddress);
278	Wire.write(regAddress);
279	Wire.endTransmission();
280	Wire.requestFrom(deviceAddress, (uint8_t)14);
281	AccelX = (((int16_t)Wire.read() << 8) Wire.read());
282	AccelY = (((int16_t)Wire.read() << 8) Wire.read());
283	AccelZ = (((int16_t)Wire.read() << 8) Wire.read());
284	Temperature = (((int16_t)Wire.read() << 8) Wire.read());
285	GyroX = (((int16_t)Wire.read() << 8) Wire.read());
286	GyroY = (((int16_t)Wire.read() << 8) Wire.read());
287	GyroZ = (((int16_t)Wire.read() << 8) Wire.read());
288	}
289	
290	//configure MPU6050
291	void MPU6050_Init() {
292	delay(150);
293	Serial.println("MPU6050 OK.");
294	I2C_Write(MPU6050SlaveAddress, MPU6050_REGISTER_SMPLRT_DIV,
295	0x07);
296	I2C_Write(MPU6050SlaveAddress, MPU6050_REGISTER_PWR_MGMT_1,
297	0x01);
298	I2C_Write(MPU6050SlaveAddress, MPU6050_REGISTER_PWR_MGMT_2,
299	0x00);
300	I2C_Write(MPU6050SlaveAddress, MPU6050_REGISTER_CONFIG, 0x00);
301	I2C_Write(MPU6050SlaveAddress, MPU6050_REGISTER_GYRO_CONFIG,
302	0x00); //set +/-250 degree/second full scale
303	I2C_Write(MPU6050SlaveAddress, MPU6050_REGISTER_ACCEL_CONFIG,
304	0x00); // set +/- 2g full scale
305	I2C_Write(MPU6050SlaveAddress, MPU6050_REGISTER_FIFO_EN, 0x00);
306	I2C_Write(MPU6050SlaveAddress, MPU6050_REGISTER_INT_ENABLE,
307	0x01);
308	I2C_Write(MPU6050SlaveAddress,
309	MPU6050_REGISTER_SIGNAL_PATH_RESET, 0x00);
310	I2C_Write(MPU6050SlaveAddress, MPU6050_REGISTER_USER_CTRL,
311	0x00);
312	}
313	
314	
315	void printDigits(int digits)
316	{
317	// utility for digital clock display: prints preceding colon and

Kode Program Perangkat Keras Sensor dan Aktuator	
318	leading 0
319	Serial.print(":");
320	if (digits < 10)
321	Serial.print('0');
322	Serial.print(digits);
323	}
324	
325	
326	
327	void digitalClockDisplay()
328	{
329	// digital clock display of the time
330	Serial.print(hour());
331	printDigits(minute());
332	printDigits(second());
333	Serial.print(" ");
334	Serial.print(day());
335	Serial.print(".");
336	Serial.print(month());
337	Serial.print(".");
338	Serial.print(year());
339	Serial.println();
340	}
341	/*----- NTP code -----*/
342	
343	const int NTP_PACKET_SIZE = 48; // NTP time is in the first 48
344	bytes of message
345	byte packetBuffer[NTP_PACKET_SIZE]; //buffer to hold incoming &
346	outgoing packets
347	
348	time_t getNtpTime()
349	{
350	IPAddress ntpServerIP; // NTP server's ip address
351	
352	while (Udp.parsePacket() > 0) ; // discard any previously
353	received packets
354	Serial.println("Transmit NTP Request");
355	// get a random server from the pool
356	WiFi.hostByName(ntpServerName, ntpServerIP);
357	Serial.print(ntpServerName);
358	Serial.print(": ");
359	Serial.println(ntpServerIP);
360	sendNTPpacket(ntpServerIP);
361	uint32_t beginWait = millis();
362	while (millis() - beginWait < 1500) {
363	int size = Udp.parsePacket();

Kode Program Perangkat Keras Sensor dan Aktuator	
364	if (size >= NTP_PACKET_SIZE) {
365	Serial.println("Receive NTP Response");
366	Udp.read(packetBuffer, NTP_PACKET_SIZE); // read packet
367	into the buffer
368	unsigned long secsSince1900;
369	// convert four bytes starting at location 40 to a long
370	integer
371	secsSince1900 = (unsigned long)packetBuffer[40] << 24;
372	secsSince1900 = (unsigned long)packetBuffer[41] << 16;
373	secsSince1900 = (unsigned long)packetBuffer[42] << 8;
374	secsSince1900 = (unsigned long)packetBuffer[43];
375	return secsSince1900 - 2208988800UL + timeZone *
376	SECS_PER_HOUR;
377	}
378	}
379	Serial.println("No NTP Response :-(");
380	return 0; // return 0 if unable to get the time
381	}
382	
383	// send an NTP request to the time server at the given address
384	void sendNTPpacket(IPAddress &address)
385	{
386	// set all bytes in the buffer to 0
387	memset(packetBuffer, 0, NTP_PACKET_SIZE);
388	// Initialize values needed to form NTP request
389	// (see URL above for details on the packets)
390	packetBuffer[0] = 0b11100011; // LI, Version, Mode
391	packetBuffer[1] = 0; // Stratum, or type of clock
392	packetBuffer[2] = 6; // Polling Interval
393	packetBuffer[3] = 0xEC; // Peer Clock Precision
394	// 8 bytes of zero for Root Delay & Root Dispersion
395	packetBuffer[12] = 49;
396	packetBuffer[13] = 0x4E;
397	packetBuffer[14] = 49;
398	packetBuffer[15] = 52;
399	// all NTP fields have been given values, now
400	// you can send a packet requesting a timestamp:
401	Udp.beginPacket(address, 123); //NTP requests are to port 123
402	Udp.write(packetBuffer, NTP_PACKET_SIZE);
403	Udp.endPacket();
404	}

A.2 KODE PROGRAM APLIKASI SMARTPHONE

ViewController.Swift

Kode Program Aplikasi <i>Smartphone</i> Halaman Utama

Kode Program Aplikasi <i>Smartphone</i> Halaman Utama	
1	//
2	// ViewController.swift
3	// SmartBag
4	//
5	// Created by admin on 11/20/17.
6	// Copyright © 2017 indosystem. All rights reserved.
7	//
8	
9	import UIKit
10	import Firebase
11	import ObjectMapper
12	import Kingfisher
13	
14	class ViewController: UIViewController {
15	@IBOutlet weak var headerView: UIView!
16	var inventories = [InventoryList]()
17	var inventoriesDictionary = [String: Inventory]()
18	var collectionViews: UICollectionView!
19	var collectionViewsTopAnchor: NSLayoutConstraint?
20	
21	@IBOutlet weak var alertNoReminder: UIView!
22	@IBOutlet weak var emptyBag: UIView!
23	
24	override func viewDidLoad() {
25	super.viewDidLoad()
26	hideKeyboardWhenTappedAround()
27	setupCollectionView()
28	observerUserInventory()
29	// Do any additional setup after loading the view,
30	typically from a nib.
31	}
32	
33	override func viewWillAppear(_ animated: Bool) {
34	observerUserInventory()
35	}
36	
37	func setupCollectionView(){
38	let layout = UICollectionViewFlowLayout()
39	layout.minimumLineSpacing = 12
40	layout.minimumInteritemSpacing = 12
41	
42	collectionViews = UICollectionView(frame: view.frame,
43	collectionViewLayout: layout)
44	collectionViews.contentInset = UIEdgeInsets(top: 22, left:
45	0, bottom: 0, right: 0)
46	collectionViews.scrollIndicatorInsets = UIEdgeInsets(top:

Kode Program Aplikasi <i>Smartphone</i> Halaman Utama	
47	0, left: 0, bottom: 0, right: 0)
48	collectionViews.showsVerticalScrollIndicator = false
49	
50	collectionViews.register(inventoryListCollectionViewCell.self,
51	forCellWithReuseIdentifier: "inventoryCell")
52	collectionViews.delegate = self
53	collectionViews.dataSource = self
54	collectionViews.backgroundColor =
55	UIColor.white.withAlphaComponent(0)
56	view.addSubview(collectionViews)
57	collectionViews.translatesAutoresizingMaskIntoConstraints
58	= false
59	collectionViews.bottomAnchor.constraint(equalTo:
60	view.bottomAnchor, constant: -20).isActive = true
61	collectionViews.widthAnchor.constraint(equalTo:
62	view.widthAnchor).isActive = true
63	collectionViews.centerXAnchor.constraint(equalTo:
64	view.centerXAnchor).isActive = true
65	collectionViewsTopAnchor =
66	collectionViews.topAnchor.constraint(equalTo:
67	headerView.bottomAnchor, constant: 10)
68	collectionViewsTopAnchor?.isActive = true
69	
70	}
71	
72	override func didReceiveMemoryWarning() {
73	super.didReceiveMemoryWarning()
74	// Dispose of any resources that can be recreated.
75	}
76	
77	override var preferredStatusBarStyle: UIStatusBarStyle{
78	return .lightContent
79	}
80	
81	func observeTodayReminder(){
82	let currentDate = Date()
83	let dateFormatter = DateFormatter()
84	dateFormatter.dateFormat = "dd-MM-yyyy"
85	let date = dateFormatter.string(from: currentDate)
86	let dateRef =
87	Database.database().reference().child("reminders")
88	dateRef.observe(.value) { (snapshot) in
89	if snapshot.hasChild(date){
90	self.alertNoReminder.isHidden = true
91	self.collectionViewsTopAnchor?.isActive = false
92	self.collectionViewsTopAnchor =

Kode Program Aplikasi <i>Smartphone</i> Halaman Utama	
93	self.collectionViews.topAnchor.constraint(equalTo:
94	self.headerView.bottomAnchor, constant: 10)
95	self.collectionViewsTopAnchor?.isActive = true
96	dateRef.child(date).observe(.childAdded, with: {
97	(snapshot) in
98	let inventoryID = snapshot.key
99	let inventoryReference =
100	Database.database().reference().child("inventory").child(inventory
101	ID)
102	
103	inventoryReference.observe(.value, with: {
104	(snapshot) in
105	let idTag = snapshot.key
106	if let lJSONArray = snapshot.value{
107	let data =
108	Mapper<Inventory>().map(JSONObject: lJSONArray)
109	if let index:Int =
110	self.inventorys.index(where: {\$0.ID == idTag}) {
111	if self.inventorys[index].status
112	== 0{
113	print("You are missing: " +
114	self.inventorys[index].nama!)
115	}else{
116	print("Found: " +
117	self.inventorys[index].nama!)
118	}
119	}else{
120	
121	self.inventorys.append(InventoryList(added: false, desc:
122	data?.deskripsi ?? "Deskripsi Barang",imgUrl: data?.imageUrl ??
123	"https://firebasestorage.googleapis.com/v0/b/smartbag-
124	b64b8.appspot.com/o/noimage.png?alt=media&token=91a48b54-6e2e-
125	43d1-8274-c66d2c679ee1",name: data?.name ?? "Nama Barang"
126	,inventID: idTag, stat: (data?.status)!, timeOut: data?.timeOut ??
127	"01-01-1980 00:00:00", timeIn: data?.timeIn ?? "01-01-1980
128	00:00:00"))
129	}
130	}
131	DispatchQueue.main.async(execute: {
132	self.collectionViews.reloadData()
133	})
134	})
135	})
136	}else{
137	self.alertNoReminder.isHidden = false
138	self.collectionViewsTopAnchor?.isActive = false

Kode Program Aplikasi <i>Smartphone</i> Halaman Utama	
139	self.collectionViewsTopAnchor =
140	self.collectionViews.topAnchor.constraint(equalTo:
141	self.alertNoReminder.bottomAnchor)
142	self.collectionViewsTopAnchor?.isActive = true
143	}
144	}
145	}
146	
147	func observerUserInventory(){
148	let ref =
149	Database.database().reference().child("inventory")
150	ref.observe(.childAdded, with: { (snapshot) in
151	let inventoryID = snapshot.key
152	let inventoryReference =
153	Database.database().reference().child("inventory").child(inventory
154	ID)
155	
156	inventoryReference.observe(.value, with: { (snapshot)
157	in
158	let idTag = snapshot.key
159	
160	if let lJSONArray = snapshot.value{
161	self.emptyBag.isHidden = true
162	let data = Mapper<Inventory>().map(JSONObject:
163	lJSONArray)
164	if let index:Int =
165	self.inventorys.index(where: {\$0.ID == idTag \$0.status == 0 &&
166	\$0.isAdded == true}) {
167	self.inventorys.remove(at: index)
168	}
169	if data?.status != 0{
170	
171	self.inventorys.append(InventoryList(added: true, desc:
172	data?.deskripsi ?? "Deskripsi Barang",imageUrl: data?.imageUrl ??
173	"https://firebasestorage.googleapis.com/v0/b/smartbag-
174	b64b8.appspot.com/o/noimage.png?alt=media&token=91a48b54-6e2e-
175	43d1-8274-c66d2c679ee1",name: data?.name ?? "Nama Barang"
176	,inventID: idTag, stat: (data?.status)!, timeOut: data?.timeOut ??
177	"01-01-1980 00:00:00", timeIn: data?.timeIn ?? "01-01-1980
178	00:00:00"))
179	}
180	if self.inventorys.isEmpty{
181	self.emptyBag.isHidden = false
182	}else{
183	self.emptyBag.isHidden = true
184	

Kode Program Aplikasi <i>Smartphone</i> Halaman Utama	
185	}
186	DispatchQueue.main.async(execute: {
187	self.collectionViews.reloadData()
188	})
189	}
190	})
191	})
192	observeTodayReminder()
193	
194	}
195	}
196	
197	
198	extension ViewController:
199	UICollectionViewDelegate, UICollectionViewDataSource,
200	UICollectionViewDelegateFlowLayout{
201	
202	//Specifying the number of sections in the Collection View
203	func numberOfSections(in collectionView: UICollectionView)->
204	Int {
205	return 1
206	}
207	//Specifying the number of cells in the Collection View
208	func collectionView(_ collectionView: UICollectionView,
209	numberOfItemsInSection section: Int) -> Int{
210	return inventorys.count
211	}
212	
213	//Method to dequeue the cell and set it up
214	func collectionView(_ collectionView: UICollectionView,
215	cellForItemAt indexPath: IndexPath) -> UICollectionViewCell{
216	let cell =
217	collectionView.dequeueReusableCell(withReuseIdentifier:
218	"inventoryCell", for: indexPath) as!
219	inventoryListCollectionViewCell
220	cell.awakeFromNib()
221	cell.delegate = self as?
222	inventoryListCollectionViewCellDelegate
223	return cell
224	}
225	
226	//Method to populate the data of a given cell
227	func collectionView(_ collectionView: UICollectionView,
228	willDisplay cell: UICollectionViewCell, forItemAt indexPath:
229	IndexPath) {
230	let inventoryyrData = inventorys[indexPath.row]

Kode	Program Aplikasi <i>Smartphone</i> Halaman Utama
231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276	<pre> let inventoryCell = cell as! inventoryListCollectionViewCell let url = URL(string: inventoryrData.imageUrl!) let dateFormatter = DateFormatter() dateFormatter.dateFormat = "dd-MM-yyyy HH:mm:ss" let dateIn = dateFormatter.date(from: inventoryrData.timeInDate!) dateFormatter.dateFormat = "HH:mm:ss" let newTimeIn = dateFormatter.string(from: dateIn!) //check if item is missing if inventoryrData.isAdded!{ inventoryCell.layer.shadowColor = UIColor(red: 226/255, green: 230/255, blue: 239/255, alpha: 1).CGColor inventoryCell.layer.shadowOffset = CGSize(width: 0, height: 1) inventoryCell.layer.shadowOpacity = 1 inventoryCell.layer.shadowRadius = 4 //Here your control your blur inventoryCell.layer.masksToBounds = false inventoryCell.layer.cornerRadius = 5 inventoryCell.backgroundColor = UIColor.white inventoryCell.missingView.isHidden = true inventoryCell.exclamationView.isHidden = true inventoryCell.exclamationLabel.isHidden = true }else{ inventoryCell.layer.shadowColor = UIColor(red: 255/255, green: 0/255, blue: 0/255, alpha: 1).CGColor inventoryCell.layer.shadowOffset = CGSize(width: 0, height: 4) inventoryCell.layer.shadowOpacity = 1 inventoryCell.layer.shadowRadius = 3 //Here your control your blur inventoryCell.layer.masksToBounds = false inventoryCell.layer.cornerRadius = 5 inventoryCell.missingView.isHidden = false inventoryCell.exclamationView.isHidden = false inventoryCell.exclamationLabel.isHidden = false } if inventoryrData.nama?.uppercased() == "NAMA BARANG"{ inventoryCell.newItemView.image = UIImage(named: "rectangleGold") inventoryCell.newItemLabel.text = "NEW ITEM !" }else{ inventoryCell.newItemView.image = nil inventoryCell.newItemLabel.text = "" </pre>

Kode Program Aplikasi <i>Smartphone</i> Halaman Utama	
277	}
278	
279	inventoryCell.imageView.kf.setImage(with: url)
280	inventoryCell.titleLabel.text = inventoryrData.nama!
281	inventoryCell.descLabel.text = inventoryrData.deskripsi
282	inventoryCell.expiryDate.text = "Time In: " + newTimeIn
283	}
284	//Set the size of cell
285	func collectionView(_ collectionView: UICollectionView, layout
286	collectionViewLayout: UICollectionViewLayout, sizeForItemAt
287	indexPath: IndexPath) -> CGSize {
288	return CGSize(width: collectionViews.frame.width / 1.15,
289	height: 147)
290	}
291	
292	func collectionView(_ collectionView: UICollectionView,
293	didSelectItemAt indexPath: IndexPath) {
294	let Storyboard = UIStoryboard(name: "Main", bundle: nil)
295	let descView =
296	Storyboard.instantiateViewController(withIdentifier:
297	"DescViewController") as! DescViewController
298	let inventData = inventorys[indexPath.row]
299	
300	descView.getID = inventData.ID
301	print(inventData.ID)
302	present(descView, animated: true, completion: nil)
303	}
304	
305	}
306	
307	extension UIViewController {
308	
309	func hideKeyboardWhenTappedAround() {
310	let tap: UITapGestureRecognizer =
311	UITapGestureRecognizer(target: self, action:
312	#selector(UIViewController.dismissKeyboard))
313	
314	
315	
316	tap.cancelsTouchesInView = false
317	view.addGestureRecognizer(tap)
318	}
319	
320	func getTopViewController() -> UIViewController?{
321	if var topController =
322	UIApplication.shared.keyWindow?.rootViewController

Kode Program Aplikasi <i>Smartphone</i> Halaman Utama	
323	{
324	while (topController.presentedViewController != nil)
325	{
326	topController =
327	topController.presentedViewController!
328	}
329	return topController
330	}
331	return nil
332	}
333	
334	func createAlert(titleText: String, messageText: String){
335	let alert = UIAlertController(title: titleText, message:
336	messageText, preferredStyle: .alert)
337	let cancelOption = UIAlertAction(title: "OK", style:
338	.default, handler: { (action) in
339	alert.dismiss(animated: true, completion: nil)
340	})
341	cancelOption.setValue(UIColor.init(red: 42/255, green:
342	147/255, blue: 137/255, alpha: 1), forKey: "titleTextColor")
343	alert.addAction(cancelOption)
344	
345	self.getTopViewController()?.present(alert, animated:
346	true, completion: nil)
347	}
348	
349	
350	@objc func dismissKeyboard(){
351	view.endEditing(true)
352	}
353	}

DescViewController.Swift

Kode Program Aplikasi <i>Smartphone</i> halaman Deskripsi Barang	
1	//
2	// DescViewController.swift
3	// SmartBag
4	//
5	// Created by admin on 03/12/17.
6	// Copyright © 2017 indosystem. All rights reserved.
7	//
8	
9	import UIKit
10	import Firebase
11	import FirebaseStorage
12	import Kingfisher

	Kode Program Aplikasi <i>Smartphone</i> halaman Deskripsi Barang
13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58	<pre> class DescViewController: UIViewController, UIImagePickerControllerDelegate, UINavigationControllerDelegate { func imagePickerController(_ picker: UIImagePickerController, didFinishPickingMediaWithInfo info: [String : Any]) { var selectedImageFromPicker: UIImage? if let editedImage = info["UIImagePickerControllerEditedImage"] as? UIImage{ selectedImageFromPicker = editedImage } else if let originalImage = info["UIImagePickerControllerOriginalImage"] as? UIImage{ selectedImageFromPicker = originalImage } if let selectedImage = selectedImageFromPicker { descImage.image = selectedImage imageChanged = true } dismiss(animated: true, completion: nil) } func imagePickerControllerDidCancel(_ picker: UIImagePickerController) { print("canceled") dismiss(animated: true, completion: nil) } @IBAction func changeImage(_ sender: Any) { let picker = UIImagePickerController() picker.delegate = self picker.allowsEditing = true present(picker, animated: true, completion: nil) } @IBOutlet weak var changeImageButton: UIButton! @IBOutlet weak var descTitle: UITextField! @IBOutlet weak var descText: UITextView! @IBOutlet weak var descImage: UIImageView! var getID: String? var imageChanged = false let rootRef = Database.database().reference() @IBAction func backButton(sender: Any) { </pre>

Kode Program Aplikasi <i>Smartphone</i> halaman Deskripsi Barang	
59	dismiss(animated: true, completion: nil)
60	}
61	
62	@IBAction func saveButton(_ sender: Any) {
63	let idRef = rootRef.child("inventory").child(getID!)
64	if imageChanged {
65	let imageName = getID
66	let storageRef =
67	Storage.storage().reference().child("images").child("\(imageName!)
68	.png")
69	if let uploadData =
70	UIImageJPEGRepresentation(self.descImage.image!, 0.1) {
71	storageRef.putData(uploadData, metadata: nil,
72	completion: { (metadata, errMsg) in
73	if errMsg != nil{
74	print(errMsg!)
75	return
76	}
77	if let imageUrl =
78	metadata?.downloadURL()?.absoluteString{
79	idRef.updateChildValues(["deskripsi" :
80	self.descText.text ?? "Deskripsi Barang"])
81	idRef.updateChildValues(["nama" :
82	self.descTitle.text?.uppercased() ?? "NAMA BARANG"])
83	idRef.updateChildValues(["imageUrl" :
84	imageUrl])
85	self.createAlert(titleText: "Sukses",
86	messageText: "Update Successful")
87	}
88	}}
89	}
90	
91	}else{
92	let idRef = rootRef.child("inventory").child(getID!)
93	idRef.updateChildValues(["deskripsi" :
94	self.descText.text ?? "Deskripsi Barang"])
95	idRef.updateChildValues(["nama" :
96	self.descTitle.text?.uppercased() ?? "NAMA BARANG"])
97	createAlert(titleText: "Sukses", messageText: "Update
98	Successful")
99	}
100	
101	}
102	
103	override var preferredStatusBarStyle: UIStatusBarStyle{
104	return .lightContent

Kode	Program Aplikasi <i>Smartphone</i> halaman Deskripsi Barang
105	}
106	
107	override func viewDidLoad() {
108	super.viewDidLoad()
109	hideKeyboardWhenTappedAround()
110	changeImageButton.layer.shadowColor =
111	UIColor.black.cgColor
112	changeImageButton.layer.shadowOffset = CGSize(width: 2.5,
113	height: 2.5)
114	changeImageButton.layer.shadowOpacity = 0.5
115	changeImageButton.layer.shadowRadius = 1
116	let idRef = rootRef.child("inventory").child(getID!)
117	idRef.observe(.value) { (data) in
118	if let dictionary = data.value as? [String: Any]{
119	
120	if dictionary["nama"] != nil{
121	self.descTitle.text = dictionary["nama"]! as?
122	String
123	}
124	
125	if dictionary["imageUrl"] != nil{
126	let url = URL(string: (dictionary["imageUrl"]!
127	as? String)!)
128	self.descImage.kf.setImage(with: url)
129	}else{
130	let url = URL(string:
131	"https://firebasestorage.googleapis.com/v0/b/smartbag-
132	b64b8.appspot.com/o/noimage.png?alt=media&token=91a48b54-6e2e-
133	43d1-8274-c66d2c679ee1")
134	self.descImage.kf.setImage(with: url)
135	}
136	
137	if dictionary["deskripsi"] != nil{
138	self.descText.text = dictionary["deskripsi"]!
139	as? String
140	}
141	
142	}
143	}
144	descText.setContentOffset(CGPoint(x: 0, y: 0), animated:
145	false)
146	descText.textContainer.lineFragmentPadding = 0
147	descText.textContainerInset = .zero
148	// Do any additional setup after loading the view.
149	}
150	

Kode Program Aplikasi <i>Smartphone</i> halaman Deskripsi Barang	
151	override func didReceiveMemoryWarning() {
152	super.didReceiveMemoryWarning()
153	// Dispose of any resources that can be recreated.
154	}
155	
156	
157	/*
158	// MARK: - Navigation
159	
160	// In a storyboard-based application, you will often want to
161	do a little preparation before navigation
162	override func prepare(for segue: UIStoryboardSegue, sender:
163	Any?) {
164	// Get the new view controller using
165	segue.destinationViewController.
166	// Pass the selected object to the new view controller.
167	}
168	*/
169	
170	}

ReminderViewController.Swift

Kode Program Aplikasi <i>Smartphone</i> Halaman <i>Reminder</i>	
1	//
2	// ReminderViewController.swift
3	// SmartBag
4	//
5	// Created by admin on 08/12/17.
6	// Copyright © 2017 indosystem. All rights reserved.
7	//
8	
9	import UIKit
10	import Firebase
11	import Kingfisher
12	import ObjectMapper
13	
14	class ReminderViewController: UIViewController {
15	
16	@IBOutlet weak var reminderTitle: UILabel!
17	@IBOutlet weak var addButton: UIButton!
18	@IBOutlet weak var dateTxtField: UITextField!
19	let rootRef = Database.database().reference()
20	
21	@IBAction func backButton(_ sender: Any) {
22	dismiss(animated: true, completion: nil)
23	}

Kode Program Aplikasi <i>Smartphone</i> Halaman <i>Reminder</i>	
24	
25	@IBAction func addTask(_ sender: Any) {
26	
27	if dateTxtField.text == ""{
28	createAlert(titleText: "Error", messageText: "Reminder
29	Date is Empty !")
30	return
31	}
32	let idRef =
33	rootRef.child("reminders").child(dateTxtField.text!)
34	if addID.isEmpty == false {
35	for element in addID{
36	idRef.updateChildValues([element : 1])
37	}
38	}
39	if removeID.isEmpty == false {
40	for element in removeID{
41	idRef.child(element).removeValue()
42	}
43	}
44	if getDate != nil{
45	createAlert(titleText: "Succes", messageText:
46	"Reminder Updated")
47	}else{
48	createAlert(titleText: "Succes", messageText:
49	"Reminder Added")
50	}
51	
52	
53	}
54	let datePicker = UIDatePicker()
55	@IBOutlet weak var whatLabel: UILabel!
56	var inventorys = [InventoryList]()
57	var inventorysDictionary = [String: Inventory]()
58	var collectionViews: UICollectionView!
59	var addID = [String]()
60	var removeID = [String]()
61	var getDate: String?
62	override func viewDidLoad() {
63	super.viewDidLoad()
64	hideKeyboardWhenTappedAround()
65	setupCollectionView()
66	editReminder()
67	observerUserInventory()
68	createDatePicker()
69	// Do any additional setup after loading the view.

Kode Program Aplikasi <i>Smartphone</i> Halaman <i>Reminder</i>	
70	}
71	
72	@objc func textFieldDidChange(_ textField: UITextField) {
73	print(textField.text)
74	}
75	override var preferredStatusBarStyle: UIStatusBarStyle{
76	return .lightContent
77	}
78	
79	override func didReceiveMemoryWarning() {
80	super.didReceiveMemoryWarning()
81	// Dispose of any resources that can be recreated.
82	}
83	
84	func setupCollectionView(){
85	let layout = UICollectionViewFlowLayout()
86	layout.minimumLineSpacing = 12
87	layout.minimumInteritemSpacing = 12
88	
89	collectionViews = UICollectionView(frame: view.frame,
90	collectionViewLayout: layout)
91	collectionViews.contentInset = UIEdgeInsets(top: 0, left:
92	0, bottom: 0, right: 0)
93	collectionViews.scrollIndicatorInsets = UIEdgeInsets(top:
94	0, left: 0, bottom: 0, right: 0)
95	collectionViews.showsVerticalScrollIndicator = false
96	
97	collectionViews.register(inventoryReminderCollectionViewCell.self,
98	forCellWithReuseIdentifier: "inventoryReminder")
99	collectionViews.delegate = self
100	collectionViews.dataSource = self
101	collectionViews.backgroundColor =
102	UIColor.white.withAlphaComponent(0)
103	view.addSubview(collectionViews)
104	collectionViews.translatesAutoresizingMaskIntoConstraints
105	= false
106	collectionViews.bottomAnchor.constraint(equalTo:
107	addButton.topAnchor, constant: -10).isActive = true
108	collectionViews.widthAnchor.constraint(equalTo:
109	view.widthAnchor).isActive = true
110	collectionViews.centerXAnchor.constraint(equalTo:
111	view.centerXAnchor).isActive = true
112	collectionViews.topAnchor.constraint(equalTo:
113	whatLabel.bottomAnchor, constant: 10).isActive = true
114	
115	}

Kode Program Aplikasi <i>Smartphone</i> Halaman <i>Reminder</i>	
116	
117	func editReminder(){
118	if getDate != nil{
119	dateTxtField.text = getDate
120	observerUserDate(date: getDate!)
121	reminderTitle.text = "UPDATE A REMINDER"
122	addButton.setTitle("EDIT REMINDER", for: .normal)
123	dateTxtField.isUserInteractionEnabled = false
124	}
125	}
126	
127	func observerUserInventory(){
128	let ref =
129	Database.database().reference().child("inventory")
130	ref.observe(.childAdded, with: { (snapshot) in
131	let inventoryID = snapshot.key
132	let inventoryReference =
133	Database.database().reference().child("inventory").child(inventory
134	ID)
135	
136	inventoryReference.observe(.value, with: { (snapshot)
137	in
138	let idTag = snapshot.key
139	if let lJSONArray = snapshot.value{
140	let data = Mapper<Inventory>().map(JSONObject:
141	lJSONArray)
142	if let index:Int =
143	self.inventorys.index(where: {\$0.ID == idTag}) {
144	self.inventorys.remove(at: index)
145	}
146	self.inventorys.append(InventoryList(added:
147	false, desc: data?.deskripsi ?? "Deskripsi Barang",imgUrl:
148	data?.imageUrl ??
149	"https://firebasestorage.googleapis.com/v0/b/smartbag-
150	b64b8.appspot.com/o/noimage.png?alt=media&token=91a48b54-6e2e-
151	43d1-8274-c66d2c679ee1",name: data?.name ?? "Nama Barang"
152	,inventID: idTag, stat: (data?.status)!, timeOut: data?.timeOut ??
153	"01-01-1980 00:00:00", timeIn: data?.timeIn ?? "01-01-1980
154	00:00:00"))
155	
156	DispatchQueue.main.async(execute: {
157	self.collectionViews.reloadData()
158	})
159	}
160	})
161	})

Kode Program Aplikasi <i>Smartphone</i> Halaman <i>Reminder</i>	
162	}
163	
164	func textFieldShouldReturn(_ textField: UITextField) -> Bool {
165	self.view.endEditing(true)
166	return false
167	}
168	
169	
170	func createDatePicker(){
171	let toolbar = UIToolbar()
172	toolbar.sizeToFit()
173	let calendar: NSCalendar = NSCalendar(calendarIdentifier:
174	NSCalendar.Identifier.gregorian)!
175	let currentDate = NSDate()
176	let dateComponents = NSDateComponents()
177	let minDate = calendar.date(byAdding: dateComponents as
178	DateComponents, to: currentDate as Date, options:
179	NSCalendar.Options(rawValue: 0))
180	
181	dateComponents.month = 3 //or you can change month =
182	day(90)
183	
184	let maxDate = calendar.date(byAdding: dateComponents as
185	DateComponents, to: currentDate as Date, options:
186	NSCalendar.Options(rawValue: 0))
187	datePicker.datePickerMode = .date
188	datePicker.backgroundColor = UIColor.white
189	datePicker.minimumDate = minDate
190	datePicker.maximumDate = maxDate
191	let doneButton = UIBarButtonItem(barButtonSystemItem:
192	.done, target: nil, action: #selector(donePressed))
193	doneButton.tintColor = UIColor.black
194	toolbar.setItems([doneButton], animated: false)
195	
196	dateTxtField.inputAccessoryView = toolbar
197	dateTxtField.inputView = datePicker
198	
199	}
200	
201	@objc func donePressed(){
202	let dateFormatter = DateFormatter()
203	dateFormatter.dateFormat = "dd-MM-yyyy"
204	dateTxtField.text = dateFormatter.string(from:
205	datePicker.date)
206	observerUserDate(date: dateTxtField.text!)
207	self.view.endEditing(true)

Kode Program Aplikasi <i>Smartphone</i> Halaman <i>Reminder</i>	
208	}
209	
210	func observerUserData(date: String){
211	let dateRef =
212	Database.database().reference().child("reminders")
213	dateRef.observe(.value) { (snapshot) in
214	if snapshot.hasChild(date){
215	print("there is reminder")
216	dateRef.child(date).observe(.childAdded, with: {
217	(snapshot) in
218	let inventoryID = snapshot.key
219	print(snapshot.value)
220	let inventoryReference =
221	Database.database().reference().child("inventory").child(inventory
222	ID)
223	
224	inventoryReference.observe(.value, with: {
225	(snapshot) in
226	let idTag = snapshot.key
227	if let lJSONArray = snapshot.value{
228	let data =
229	Mapper<Inventory>().map(JSONObject: lJSONArray)
230	print(lJSONArray)
231	if let index:Int =
232	self.inventorys.index(where: {\$0.ID == idTag}) {
233	self.inventorys[index].isAdded =
234	true
235	}
236	DispatchQueue.main.async(execute: {
237	self.collectionViews.reloadData()
238	})
239	}
240	})
241	})
242	}else{
243	for element in self.inventorys{
244	element.isAdded = false
245	}
246	DispatchQueue.main.async(execute: {
247	self.collectionViews.reloadData()
248	})
249	}
250	}
251	
252	}
253	

Kode Program Aplikasi <i>Smartphone</i> Halaman <i>Reminder</i>	
254	/*
255	// MARK: - Navigation
256	
257	// In a storyboard-based application, you will often want to
258	do a little preparation before navigation
259	override func prepare(for segue: UIStoryboardSegue, sender:
260	Any?) {
261	// Get the new view controller using
262	segue.destinationViewController.
263	// Pass the selected object to the new view controller.
264	}
265	*/
266	
267	}
268	
269	extension ReminderViewController:
270	UICollectionViewDelegate, UICollectionViewDataSource,
271	UICollectionViewDelegateFlowLayout{
272	
273	//Specifying the number of sections in the Collection View
274	func numberOfSections(in collectionView: UICollectionView)->
275	Int {
276	return 1
277	}
278	//Specifying the number of cells in the Collection View
279	func collectionView(_ collectionView:UICollectionView,
280	numberOfItemsInSection section: Int) -> Int{
281	return inventorys.count
282	}
283	
284	//Method to dequeue the cell and set it up
285	func collectionView(_ collectionView:UICollectionView,
286	cellForItemAt indexPath: IndexPath) -> UICollectionViewCell{
287	let cell =
288	collectionView.dequeueReusableCell(withReuseIdentifier:
289	"inventoryReminder", for: indexPath) as!
290	inventoryReminderCollectionViewCell
291	cell.awakeFromNib()
292	cell.delegate = self as?
293	inventoryReminderCollectionViewCellDelegate
294	return cell
295	}
296	
297	//Method to populate the data of a given cell
298	func collectionView(_ collectionView: UICollectionView,
299	willDisplay cell: UICollectionViewCell, forItemAt indexPath:

Kode Program Aplikasi <i>Smartphone</i> Halaman <i>Reminder</i>	
300	IndexPath) {
301	let inventoryrData = inventories[indexPath.row]
302	let inventoryCell = cell as!
303	inventoryReminderCollectionViewCell
304	let url = URL(string: inventoryrData.imageUrl!)
305	if inventoryrData.isAdded!{
306	inventoryCell.button.backgroundColor = UIColor.red
307	inventoryCell.button.setTitle("Remove", for: .normal)
308	inventoryCell.buttonChanged = true
309	}else if !inventoryrData.isAdded!{
310	inventoryCell.button.backgroundColor = UIColor.green
311	inventoryCell.button.setTitle("Add", for: .normal)
312	inventoryCell.buttonChanged = false
313	}
314	inventoryCell.ID = inventoryrData.ID
315	inventoryCell.imageView.kf.setImage(with: url)
316	inventoryCell.titleLabel.text = inventoryrData.nama!
317	inventoryCell.descLabel.text = inventoryrData.deskripsi
318	}
319	//Set the size of cell
320	func collectionView(_ collectionView: UICollectionView, layout
321	collectionViewLayout: UICollectionViewLayout, sizeForItemAt
322	indexPath: IndexPath) -> CGSize {
323	return CGSize(width: collectionViews.frame.width / 1.15,
324	height: 147)
325	}
326	
327	func collectionView(_ collectionView: UICollectionView,
328	didSelectItemAt indexPath: IndexPath) {
329	let inventData = inventories[indexPath.row]
330	let date = Date()
331	let formatter = DateFormatter()
332	formatter.dateFormat = "dd-MM-yyyy"
333	let result = formatter.string(from: date)
334	print(result)
335	print(inventData.ID!)
336	}
337	
338	}
339	
340	extension ReminderViewController:
341	inventoryReminderCollectionViewCellDelegate {
342	func changeColorOfButton(forCell:
343	inventoryReminderCollectionViewCell){
344	let ID = forCell.ID
345	if let index:Int = self.addID.index(where: {\$0 == ID}) {

Kode Program Aplikasi <i>Smartphone</i> Halaman <i>Reminder</i>	
346	self.addID.remove(at: index)
347	}
348	if let index:Int = self.removeID.index(where: {\$0 == ID})
349	{
350	self.removeID.remove(at: index)
351	}
352	if !forCell.buttonChanged{
353	forCell.button.backgroundColor = UIColor.red
354	forCell.button.setTitle("Remove", for: .normal)
355	forCell.buttonChanged = !forCell.buttonChanged
356	self.addID.append(ID!)
357	if let index:Int = self.inventorys.index(where: {\$0.ID
358	== ID}) {
359	self.inventorys[index].isAdded = true
360	}
361	}else{
362	forCell.button.backgroundColor = UIColor.green
363	forCell.button.setTitle("Add", for: .normal)
364	forCell.buttonChanged = !forCell.buttonChanged
365	self.removeID.append(ID!)
366	if let index:Int = self.inventorys.index(where: {\$0.ID
367	== ID}) {
368	self.inventorys[index].isAdded = false
369	}
370	}
371	}
372	}

ReminderListViewController.Swift

Kode Program Aplikasi <i>Smartphone</i> Halaman <i>List Reminder</i>	
1	//
2	// ReminderListViewController.swift
3	// SmartBag
4	//
5	// Created by admin on 10/12/17.
6	// Copyright © 2017 indosystem. All rights reserved.
7	//
8	
9	import UIKit
10	import Firebase
11	import ObjectMapper
12	
13	
14	class ReminderListViewController: UIViewController {
15	@IBOutlet weak var headerView: UIView!
16	@IBOutlet weak var emptyReminder: UIView!

Kode Program Aplikasi <i>Smartphone</i> Halaman <i>List Reminder</i>	
17	var collectionViews: UICollectionView!
18	var reminders = [ReminderList]()
19	var reminderDates = [String]()
20	let rootRef = Database.database().reference()
21	
22	@IBOutlet weak var addButton: UIButton!
23	override func viewDidLoad() {
24	super.viewDidLoad()
25	// observerUserReminder()
26	setupCollectionView()
27	}
28	override func viewWillAppear(_ animated: Bool) {
29	observerUserReminder()
30	}
31	
32	func setupCollectionView(){
33	let layout = UICollectionViewFlowLayout()
34	layout.minimumLineSpacing = 12
35	layout.minimumInteritemSpacing = 12
36	
37	collectionViews = UICollectionView(frame: view.frame,
38	collectionViewLayout: layout)
39	collectionViews.contentInset = UIEdgeInsets(top: 10, left:
40	0, bottom: 0, right: 0)
41	collectionViews.scrollIndicatorInsets = UIEdgeInsets(top:
42	0, left: 0, bottom: 0, right: 0)
43	collectionViews.showsVerticalScrollIndicator = false
44	
45	collectionViews.register(reminderListCollectionViewCell.self,
46	forCellWithReuseIdentifier: "reminderCell")
47	collectionViews.delegate = self
48	collectionViews.dataSource = self
49	collectionViews.backgroundColor =
50	UIColor.white.withAlphaComponent(0)
51	view.addSubview(collectionViews)
52	collectionViews.translatesAutoresizingMaskIntoConstraints
53	= false
54	collectionViews.bottomAnchor.constraint(equalTo:
55	addButton.topAnchor).isActive = true
56	collectionViews.widthAnchor.constraint(equalTo:
57	view.widthAnchor).isActive = true
58	collectionViews.centerXAnchor.constraint(equalTo:
59	view.centerXAnchor).isActive = true
60	collectionViews.topAnchor.constraint(equalTo:
61	headerView.bottomAnchor, constant: 10).isActive = true
62	

Kode Program Aplikasi <i>Smartphone</i> Halaman <i>List Reminder</i>	
63	}
64	
65	override var preferredStatusBarStyle: UIStatusBarStyle{
66	return .lightContent
67	}
68	
69	override func didReceiveMemoryWarning() {
70	super.didReceiveMemoryWarning()
71	// Dispose of any resources that can be recreated.
72	}
73	override func viewWillAppear(_ animated: Bool) {
74	//rootRef.removeAllObservers()
75	}
76	
77	func observerUserReminder(){
78	//check if there is no reminder
79	rootRef.observe(.value) { (snapshot) in
80	if !snapshot.hasChild("reminders"){
81	self.emptyReminder.isHidden = false
82	}
83	DispatchQueue.main.async(execute: {
84	self.collectionViews.reloadData()
85	})
86	}
87	reminders.removeAll()
88	let ref = rootRef.child("reminders")
89	ref.observe(.childAdded, with: { (snapshot) in
90	let reminderDate = snapshot.key
91	let reminderReference =
92	self.rootRef.child("reminders").child(reminderDate)
93	reminderReference.observe(.childAdded, with: {
94	(snapshot) in
95	let inventoryID = snapshot.key
96	let inventoryReference =
97	self.rootRef.child("inventory").child(inventoryID)
98	
99	inventoryReference.observe(.value, with: {
100	(snapshot) in
101	if let lJSONArray = snapshot.value{
102	let data =
103	Mapper<Reminder>().map(JSONObject: lJSONArray)
104	self.reminderDates.append((data?.name)!)
105	if let index:Int =
106	self.reminders.index(where: {\$0.dates == reminderDate}) {
107	
108	self.reminders[index].nama?.append((data?.name)!)

Kode Program Aplikasi <i>Smartphone</i> Halaman <i>List Reminder</i>	
109	}else{
110	
111	self.reminders.append(ReminderList(date: reminderDate, name:
112	self.reminderDates))
113	}
114	self.reminderDates.removeAll()
115	}
116	if self.reminders.isEmpty{
117	self.emptyReminder.isHidden = false
118	}else{
119	self.emptyReminder.isHidden = true
120	
121	}
122	DispatchQueue.main.async(execute: {
123	self.collectionViews.reloadData()
124	})
125	})
126	
127	})
128	
129	})
130	}
131	
132	}
133	
134	extension ReminderListViewController:
135	UICollectionViewDelegate, UICollectionViewDataSource,
136	UICollectionViewDelegateFlowLayout{
137	
138	//Specifying the number of sections in the Collection View
139	func numberOfSections(in collectionView: UICollectionView)->
140	Int {
141	return 1
142	}
143	//Specifying the number of cells in the Collection View
144	func collectionView(_ collectionView:UICollectionView,
145	numberOfItemsInSection section: Int) -> Int{
146	return reminders.count
147	}
148	
149	//Method to dequeue the cell and set it up
150	func collectionView(_ collectionView:UICollectionView,
151	cellForItemAt indexPath: IndexPath) -> UICollectionViewCell{
152	let cell =
153	collectionView.dequeueReusableCell(withReuseIdentifier:
154	"reminderCell", for: indexPath) as! reminderListCollectionViewCell

Kode Program Aplikasi <i>Smartphone</i> Halaman <i>List Reminder</i>	
155	cell.awakeFromNib()
156	cell.delegate = self as?
157	reminderListCollectionViewCellDelegate
158	return cell
159	}
160	
161	//Method to populate the data of a given cell
162	func collectionView(_ collectionView: UICollectionView,
163	willDisplay cell: UICollectionViewCell, forItemAt indexPath:
164	IndexPath) {
165	let reminderData = reminders[indexPath.row]
166	let reminderCell = cell as! reminderListCollectionViewCell
167	let dateFormatter = DateFormatter()
168	dateFormatter.dateFormat = "dd-MM-yyyy"
169	let reminderDate = dateFormatter.date(from:
170	reminderData.dates!)
171	dateFormatter.dateFormat = "dd MMMM yyyy"
172	let newDate = dateFormatter.string(from: reminderDate!)
173	
174	reminderCell.descLabel.text = ""
175	if reminderData.nama?.count == 1{
176	reminderCell.descLabel.text = "• " +
177	(reminderData.nama?[0].capitalized)!
178	}else{
179	for element in reminderData.nama!{
180	reminderCell.descLabel.text =
181	reminderCell.descLabel.text! + "• " + element.capitalized + "\n"
182	}
183	}
184	
185	reminderCell.date = reminderData.dates
186	reminderCell.titleLabel.text = newDate
187	
188	}
189	//Set the size of cell
190	func collectionView(_ collectionView: UICollectionView, layout
191	collectionViewLayout: UICollectionViewLayout, sizeForItemAt
192	indexPath: IndexPath) -> CGSize {
193	let reminderData = reminders[indexPath.row]
194	return CGSize(width: collectionViews.frame.width / 1.15,
195	height: CGFloat(30 + (20 * (reminderData.nama?.count)!)))
196	}
197	
198	func collectionView(_ collectionView: UICollectionView,
199	didSelectItemAt indexPath: IndexPath) {
200	

Kode	Program Aplikasi <i>Smartphone</i> Halaman <i>List Reminder</i>
201	}
202	
203	}
204	
205	extension ReminderListViewController:
206	reminderListCollectionViewCellDelegate {
207	
208	func deleteButton(forCell: reminderListCollectionViewCell) {
209	let idRef =
210	rootRef.child("reminders").child(forCell.date!)
211	idRef.removeValue()
212	observerUserReminder()
213	}
214	
215	func editButton(forCell: reminderListCollectionViewCell) {
216	let Storyboard = UIStoryboard(name: "Main", bundle: nil)
217	let descView =
218	Storyboard.instantiateViewController(withIdentifier:
219	"ReminderViewController") as! ReminderViewController
220	descView.getDate = forCell.date
221	present(descView, animated: true, completion: nil)
222	}
223	
224	}